Mikrokopter VSM User Guide

UgCS 2.5.242

Copyright © 2015, Smart Projects Holdings Ltd

Contents

1 MikroKopter VSM User Guide **1**

1.1 First time vehicle connection . . . 1

1.2 Mission execution specifics . . . 2

1.2.1 Camera trigger action . . . 2

1.3 Command execution specifics . . . 3

1.4 Telemetry information specifics . . . 3

1.5 Fail-safe actions . . . 3

1.6 Configuration file . . . 3

1.6.1 WP-event value . . . 4

1.6.2 Common parameters . . . 4

1.6.3 Serial port configuration . . . 4

1.7 Common configuration file parameters . . . 4

1.7.1 UgCS server configuration . . . 4

1.7.2 Logging configuration . . . 5

1.7.3 Serial port configuration . . . 5

1.7.4 Network connection configuration . . . 6

1.7.5 Mission dump path . . . 7

2 Disclaimer **8**

Copyright © 2015, Smart Projects Holdings Ltd

1 MikroKopter VSM User Guide

1.1 First time vehicle connection

See Disclaimer.

Please follow these steps to connect an MikroKopter vehicle to the U[g]CS:

1. MikroKopter vehicle must be properly configured, calibrated and tested using tools and instruction from the official `MikroKopter web site` prior to using it with U[g]CS. U[g]CS does not support initial configuration, setup and calibration of MikroKopter vehicles.
2. Turn on the vehicle and plug in the radio modem paired with the vehicle or direct USB cable from the MikroKopter board to the computer where VSM is running. U[g]CS uses serial ports for communication with MikroKopter vehicles. Serial interface based communication devices like WI232 radio modems (and their analogs) and direct USB-serial connections are supported, as long as OS driver for virtual serial port is installed and serial port is successfully created. Please refer to your communication equipment manufacturer documentation about driver installation instructions.
3. Open *Vehicles* window in U[g]CS Client and wait until new vehicle appears there automatically. Both Uplink and Downlink connections should be available. Press *Gain control* and *Edit* to select corresponding vehicle profile and change the default vehicle name to be convenient for you:

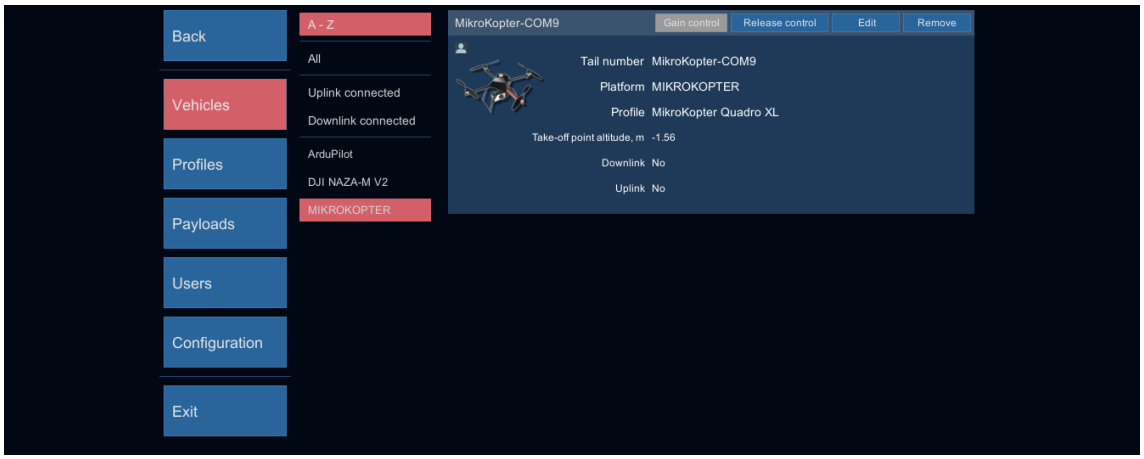

Figure 1: New MikroKopter vehicle

   Vehicle profile needs to be assigned to allow mission planning with this vehicle. Image needs to be assigned to see vehicle location on the map.

4. Repeat steps above for each your MikroKopter vehicle.

Please note that there is no technical possibility to distinguish between different MK drones if they are connected via the same port. The only way to identify drones is looking at their serial port instances. To be sure that you have permanent drones mapping, please ensure that you connect each drone via a dedicated serial port.

Copyright © 2015, Smart Projects Holdings Ltd

This firstly means that you cannot share the same radio-modem or USB-serial cable between different drones. If you are using Windows OS it usually reserves unique serial port number for each USB-serial device. On Linux OS you will need to take additional measures - manually provide udev rules for permanent device mapping based on its serial number (if such exists. Some USB devices do not have serial number specified, you will not be able to use them to connect multiple MikroKopters). Udev rules creation example: http://noctis.-de/ramblings/linux/49-howto-fixed-name-for-a-udev-device.html

If you use your own serial device naming on Linux, do not forget to add the corresponding name to the VSM configuration:

```
vehicle.mikrokopter.serial_port.2.name = /dev/mikrokopter_1
vehicle.mikrokopter.serial_port.2.baud.1 = 57600
```

Supported MikroKopter firmware versions:

- 0.x
- 2.00 may be supported but not tested.

1.2 Mission execution specifics

Before executing a mission, Mikrokopter must be set to Altitude hold mode and throttle must be in mid position!

Flight plan element / action	Support	Notes
Camera control	Partial	Only pitch control is supported.
Camera trigger	Partial	Only photo shot is supported.
Panorama	Partial	Panorama is always counted from North or last set heading
Take-off	No	
Landing	No	
Set camera by time	No	
Set camera by distance	No	

Take-off and landing is not supported in scope of automatic mission execution. You should take-off or land the drone either manually or using auto-start/landing feature of the MikroKopter which can be activated by switch on RC. See MikroKopter vendor documentation.

CareFree feature must be enabled in MikroKopter for all actions which automatically control copter heading (like POI, yaw set etc.) See http://www.mikrokopter.de/ucwiki/en/CareFree

None of fail-safe mode adjustments is supported in the U[g]CS mission parameters. Use MikroKopter original software instead to set up fail-safe parameters.

1.2.1 Camera trigger action

Can be used to trigger payload-specific action during mission execution. It triggers WP-event in the Mikro-Kopter. See http://www.mikrokopter.de/ucwiki/en/WaypointEvent and http://www.-mikrokopter.de/ucwiki/en/WpEvent for more detailed description of this MK feature. As described there, before you can use this feature you should set up pattern for the flight controller output OUT1 or SRV3 using KopterTool software. You may want to use shutter cable or IR-controller provided by the drone manufacturer.

When the "take photo" action is fired the waypoint is created with wp-event-channel field set to the value specified in the VSM configuration (see WP-event value section). It is duration in deciseconds of one unit in output signal pattern. Also waypoint duration is set to time which is enough to execute at least one cycle of the pattern. Due to granularity difference in MK protocol the pattern may start the second cycle of execution before the waypoint is finished.

Copyright © 2015, Smart Projects Holdings Ltd

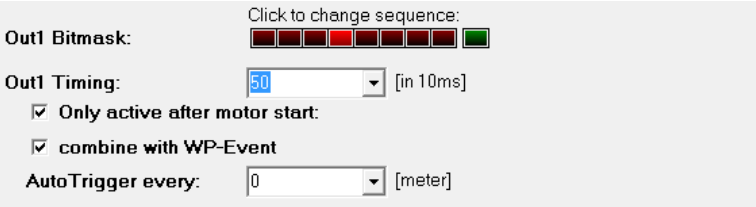

Figure 2: Example of output configuration

In KopterTool output configuration ensure you have set "Combine with WP-event" option.

1.3 Command execution specifics

Command	Support	Notes
ARM	No	
DISARM	No	
AUTOMODE	No	
MANUALMODE	No	
RETURNHOME	Yes	Current mission is erased in the device. It should be uploaded again in order to run it. Works only when the drone is in AUTO mode.
TAKEOFF	No	
LAND	No	
EMERGENCYLAND	No	

Use your RC to execute commands and switch between flight modes according to the manufacturer instructions.

1.4 Telemetry information specifics

Nothing specific.

1.5 Fail-safe actions

Fail-safe actions can be set only in KopterTool.

1.6 Configuration file

Default configuration file of the MikroKopter VSM suits most needs and it is generally not necessary to modify it.

Configuration file location:

- **On Microsoft Windows:**

```
C:\Program Files (x86)\UgCS\bin\vsm-mikrokopter.conf
```

- **On GNU/Linux:**

```
/etc/opt/ugcs/vsm-mikrokopter.conf
```

- **On Apple OS X:**

```
/Users/[user name]/Library/Application Support/UGCS/configuration/vsm-mikrokopter.conf
```

Copyright © 2015, Smart Projects Holdings Ltd

1.6.1 WP-event value

This parameter defines which value will be sent in wp-event-value field of the waypoint with "take photo" action. See Camera trigger action section.

```
vehicle.mikrokopter.wp_event_value = 50
```

1.6.2 Common parameters

All VSMs share a common set of configuration file parameters described in Common configuration file parameters. MikroKopter VSM configuration file prefix is:

```
vehicle.mikrokopter
```

1.6.3 Serial port configuration

Mandatory. At least one serial port should be configured, otherwise VSM will not try to connect to the vehicle.

- **Name:** vehicle.mikrokopter.serial_port
- **Description:** Serial port configuration, for more details see Serial port configuration. Default MikroKopter serial port communication speed is 57600 baud.
- **Example:**

```
vehicle.mikrokopter.serial_port.1.name = com1
vehicle.mikrokopter.serial_port.1.baud = 57600
```

1.7 Common configuration file parameters

VSM configuration file is a text file specified via command line argument - *-config* of the VSM application. Example:

```
--config /etc/opt/ugcs/vsm-ardupilot.conf
```

Each configuration parameter is defined as a line in the configuration file with the following structure:

```
name1.name2....nameX = value
```

where name1, name2 ... nameX are arbitrary names separated by dots to divide a variable into logical blocks and a value which can be a number value or a text string depending on the context. See below the description about common VSM configuration parameters.

1.7.1 UgCS server configuration

1.7.1.1 Listening address

Mandatory.

- **Name:** ucs.local_listening_address = [IP address]
- **Description:** Local TCP address to listen for incoming connections from UgCS server. Specify *0.0.0.0* to listen from all local addresses.
- **Example:** ucs.local_listening_address = 0.0.0.0

Copyright © 2015, Smart Projects Holdings Ltd

1.7.1.2 Listening port

Mandatory.

- **Name:** ucs.local_listening_port = [port number]
- **Description:** Local TCP port to listen for incoming connections from UgCS server. Default is 5556.
- **Example:** ucs.local_listening_port = 5556

1.7.2 Logging configuration

1.7.2.1 Level

Optional.

- **Name:** log.level = [error|warning|info|debug]
- **Description:** Logging level.
- **Default:** info
- **Example:** log.level = debug

1.7.2.2 File path

Optional.

- **Name:** log.file_path = [path to a file]
- **Description:** Absolute or relative (to the current directory) path to a logging file. Logging is disabled if logging file is not defined. File should be writable. Backslash should be escaped with a backslash.
- **Example:** log.file = /var/opt/ugcs/log/vsm-ardupilot/vsm-ardupilot.log
- **Example:** log.file = C:\\Users\\John\\AppData\\Local\\UGCS\\logs\\vsm-ardupilot\\vsm-ardupilot.log

1.7.2.3 Maximum single file size

Optional.

- **Name:** log.single_max_size = [size]
- **Description:** Maximum size of a single log file. When maximum size is exceeded, existing file is renamed by adding a time stamp and logging is continued into the empty file. [size] should be defined as a number postfixed by a case insensitive multiplier:
  - Gb, G, Gbyte, Gbytes: for Giga-bytes
  - Mb, M, Mbyte, Mbytes: for Mega-bytes
  - Kb, K, Kbyte, Kbytes: for Kilo-bytes
  - no postfix: for bytes
- **Default:** 100 Mb
- **Example:** log.single_max_size = 500 Mb

1.7.3 Serial port configuration

Optional. VSM which communicates with vehicles via serial ports should define at least one serial port, otherwise VSM will not try to connect to the vehicles. Port name and baud rate should be both defined. [prefix] is unique for each VSM.

Copyright © 2015, Smart Projects Holdings Ltd

1.7.3.1 Port name

Optional.

- **Name:** [prefix].[port index].name = [regular expression]
- **Description:** Ports which should be used to connect to the vehicles by given VSM. Port names are defined by a [regular expression] which can be used to define just a single port or create a port filtering regular expression. Expression is case insensitive on Windows. [port index] is a arbitrary port indexing name.
- **Example:** vehicle.ardupilot.serial_port.1.name = /dev/ttyUSB[0-9]+|com[0-9]+
- **Example:** vehicle.ardupilot.serial_port.2.name = com42

1.7.3.2 Port baud rate

Optional.

- **Name:** [prefix].[port index].baud.[baud index] = [baud]
- **Description:** Baud rate for port opening. [baud index] is an optional arbitrary name used when it is necessary to open the same serial port using multiple baud rates. [port index] is an arbitrary port indexing name.
- **Example:** vehicle.ardupilot.serial_port.1.baud.1 = 9600
- **Example:** vehicle.ardupilot.serial_port.1.baud.2 = 57600
- **Example:** vehicle.ardupilot.serial_port.2.baud = 38400

1.7.3.3 Excluded port name

Optional.

- **Name:** [prefix].exclude.[exclude index] = [regular expression]
- **Description:** Ports which should not be used for vehicle access by this VSM. Port names are defined by a [regular expression] which can be used to define just a single port or create a port filtering regular expression. Filter is case insensitive on Windows. [exclude index] is a arbitrary indexing name used when more than one exclude names are defined.
- **Example:** vehicle.ardupilot.serial_port.exclude.1 = /dev/ttyS.∗
- **Example:** vehicle.ardupilot.serial_port.exclude = com1

1.7.3.4 Serial port arbiter

Optional.

- **Name:** [prefix].use_serial_arbiter = [yes|no]
- **Description:** Enable (yes) or disable (no) serial port access arbitration between VSMs running on the same machine. It is recommended to have it enabled to avoid situation when multiple VSMs try to open the same port simultaneously.
- **Default:** yes
- **Example:** vehicle.ardupilot.serial_port.use_serial_arbiter = no

1.7.4 Network connection configuration

Optional. VSM which communicates with vehicles via network should define at least one network connection, otherwise VSM will not try to connect to vehicles. [prefix] is unique for each VSM.

Copyright © 2015, Smart Projects Holdings Ltd

1.7.4.1 Local IP-address for UDP

Optional.

- **Name:** [prefix].detector.[con index].udp_local_address = [IP-address]
- **Description:** Local IP-address to listen for incoming UDP packets on. Specify 0.0.0.0 if you want to listen on all local addresses.
- **Example:** vehicle.ardrone.detector.1.udp_local_address = 0.0.0.0

1.7.4.2 Local UDP port

Optional.

- **Name:** [prefix].detector.[con index].udp_local_port = [port number]
- **Description:** Local UDP port to listen for incoming packets on.
- **Example:** vehicle.ardrone.detector.1.udp_local_port = 14550

1.7.4.3 Remote IP-address for UDP

Optional.

- **Name:** [prefix].detector.[con index].udp_address = [IP-address]
- **Description:** Remote IP-address to send outgoing UDP packets to.
- **Example:** vehicle.ardrone.detector.1.udp_address = 192.168.1.1

1.7.4.4 Remote UDP port

Optional.

- **Name:** [prefix].detector.[con index].udp_port = [port number]
- **Description:** Remote UDP port to send outgoing packets to.
- **Example:** vehicle.ardrone.detector.1.udp_port = 14551

1.7.5 Mission dump path

Optional.

- **Name:** [prefix].mission_dump_path = [path to a file]
- **Description:** File to dump all generated missions to. Timestamp is appended to the name. Delete the entry to disable mission dumping. All directories in the path to a file should be already created.
- **Example:** vehicle.ardupilot.mission_dump_path = C:\\tmp\\ardupilot_dump

Copyright © 2015, Smart Projects Holdings Ltd

2 Disclaimer

DISCLAIMER OF WARRANTIES AND LIMITATIONS ON LIABILITY.

(a) SMART PROJECTS HOLDINGS LTD MAKE NO REPRESENTATIONS OR WARRANTIES REGARDING T-HE ACCURACY OR COMPLETENESS OF ANY CONTENT OR FUNCTIONALITY OF THE PRODUCT AND ITS DOCUMENTATION.

(b) SMART PROJECTS HOLDINGS LTD DISCLAIM ALL WARRANTIES IN CONNECTION WITH THE PRODUCT, AND WILL NOT BE LIABLE FOR ANY DAMAGE OR LOSS RESULTING FROM YOUR USE OF THE PRODUCT. INCLUDING BUT NOT LIMITED TO INJURY OR DEATH OF USER OR ANY THIRD PERSONS OR DAMAGE TO PROPERTY.

(c) THE SOFTWARE IS SUPPLIED AS IS WITH NO WARRANTIES AND CAN BE USED ONLY AT USERS OWN RISK.

Copyright © 2015, Smart Projects Holdings Ltd